CS10720 Problems and Solutions

Thomas Jansen

Today: Boolean Logic, Truth Tables and Digital Circuits

February 11th

Announcements

Remember deadline for portfolio submission Friday, 7pm
containing lecture summary only for today
and answers to **both** questions for one practicals problem

Announcement no feedback on Monday
but later next week (and I don't know yet when)

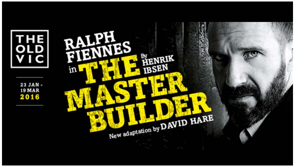
But why?

Fact marking and feedback takes about 7 minutes per student
adding up to about 18 hours

But that's what a lecturer's weekend is for!

Well... normally that's what I do but this weekend...

Saturday **Sunday**



Plans for Today

A Tiny Bit of Logic and Algebra...

Propositional logic

- proposition is a statement that is either true or false
  (Remark independent of whether you know)
- two propositions can be combined using the operators and and or to form a new one
  - when using and, new proposition true if both propositions true
  - when using or, new proposition true if at least one of the propositions true
- using not new proposition can be formed using one proposition
  new proposition is true if old proposition is false

Definition by truth tables for propositions A, B

and (conjunction)			or (disjunction)			not (negation)	
A	B	A and B	A	B	A or B	A	not A
false	false	false	false	false	false	false	true
false	true	false	false	true	true	true	false
true	false	false	true	false	true		
true	true	true	true	true	true		

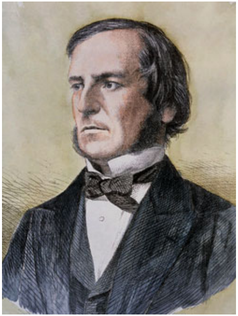
Boolean Algebra

Definition

Let $B = \{0, 1\}$. For $x, y \in B$, let $x \cap y = \min\{x, y\}$. For $x, y \in B$, let $x \cup y = \max\{x, y\}$. For $x \in B$, let $\overline{x} = 1 - x$. $(B, \cup, \cap, \bar{\ })$ is called Boolean algebra.

Observe

x	y	$x \cap y$	x	y	$x \cup y$	x	$\overline{x}$
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		



George Boole (1815–1864)

Observe close relationship between propositional logic and Boolean algebra

Notation we use interchangeably
'true' and '1', 'false' and '0'
'x and y', '$x \cap y$', '$x \wedge y$', '$x \cdot y$', 'xy'
'x or y', '$x \cup y$', '$x \vee y$'

More Operations

Conjunction (and)			Disjunction (or)			Negation (not)	
x	y	$x \wedge y$	x	y	$x \vee y$	x	$\overline{x}$
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Implication (if)			Equivalence (iff)		
x	y	$x \Rightarrow y$	x	y	$x \Leftrightarrow y$
0	0	1	0	0	1
0	1	1	0	1	0
1	0	0	1	0	0
1	1	1	1	1	1

Exclusive Or (xor)			NAND (not and)		
x	y	$x \oplus y$	x	y	$\overline{x \wedge y}$
0	0	0	0	0	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	1	1	0

'Computing' with Boolean Logic

- commutativity
  $(x \vee y) = (y \vee x)$ (also for $\wedge$, $\oplus$, $\Leftrightarrow$)
- associativity
  $((x \wedge y) \wedge z) = (x \wedge (y \wedge z))$ (also for $\vee$, $\oplus$, $\Leftrightarrow$)
- distributivity
  $(x \vee (y \wedge z)) = ((x \vee y) \wedge (x \vee z))$
  $(x \wedge (y \vee z)) = ((x \wedge y) \vee (x \wedge z))$
- elimination of double negation
  $(\neg (\neg x)) = x$
- De Morgan
  $(\overline{x \wedge y}) = (\overline{x} \vee \overline{y})$, $(\overline{x \vee y}) = (\overline{x} \wedge \overline{y})$
- elimination of implication
  $(x \Rightarrow y) = (\overline{x} \vee y)$
- elimination of equivalence
  $(x \Leftrightarrow y) = ((x \Rightarrow y) \wedge (y \Rightarrow x))$

Proofs by truth tables (i. e., complete enumeration)

One Example Proof: De Morgan

(well, two actually...)

x	y	$\overline{x \wedge y}$	$\overline{x} \vee \overline{y}$	equal?
0	0	1	1	✓
0	1	1	1	✓
1	0	1	1	✓
1	1	0	0	✓

proves $(\overline{x \wedge y}) = (\overline{x} \vee \overline{y})$

x	y	$\overline{x \vee y}$	$\overline{x} \wedge \overline{y}$	equal?
0	0	1	1	✓
0	1	0	0	✓
1	0	0	0	✓
1	1	0	0	✓

proves $(\overline{x \vee y}) = (\overline{x} \wedge \overline{y})$

Boolean Functions

Definition function $f\colon B^n \to B$ (Remember $f\colon \{0,1\}^n \to \{0,1\}$)
is called Boolean function over n variables

Example $f\colon B^3 \to B$

x	y	z	$f(x,y,z)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Truth table bulky. Are there more compact representations?

Observation if we agree on order then value vector suffices
$(0,1,0,0,0,1,0,1)$

Can we find a Boolean expression for this?

Minterms

Definition conjunction of all n variables (either negated or unnegated) is called minterm for these n variables

Example minterm $x\,\overline{y}\,z$

x	y	z	$x\,\overline{y}\,z$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Observation minterm 1 for exactly one input and 0 everywhere else
holds for every minterm

Consequence using disjunction of minterms
any Boolean function can be easily represented

Disjunctive Normal Forms

Remember example $f\colon B^3 \to B$

x	y	z	$f(x,y,z)$	$\overline{x}\,\overline{y}\,z$	$x\,\overline{y}\,z$	$x\,y\,z$
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	1	0	1	0
1	1	0	0	0	0	0
1	1	1	1	0	0	1

Consequence $f(x,y,z) = (\overline{x}\,\overline{y}\,z) \vee (x\,\overline{y}\,z) \vee (x\,y\,z)$

Definition representation of Boolean function f
by disjunction of minterms
is called disjunctive normal form

Truth Tables for Boolean Functions

How can we compute a truth table for a complex Boolean function like $f(w,x,y,z) = (w \vee x) \oplus ((w \wedge y) \vee (w \wedge (\overline{x \vee z})))$?

Idea construct truth table sub-expression by sub-expression

w	x	y	z	$\overbrace{w \vee x}^{t_1}$	$\overbrace{w \wedge y}^{t_2}$	$\overbrace{\overline{x \vee z}}^{t_3}$	$\overbrace{w \wedge t_3}^{t_4}$	$\overbrace{t_2 \vee t_4}^{t_5}$	$\overbrace{t_1 \oplus t_5}^{f}$	$f(w,x,y,z)$
0	0	0	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	1	1
0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	1	0	0	0	0	1	1
0	1	1	1	1	0	0	0	0	1	1
1	0	0	0	1	0	1	1	1	0	0
1	0	0	1	1	0	0	0	0	1	1
1	0	1	0	1	1	1	1	1	0	0
1	0	1	1	1	1	0	0	1	0	0
1	1	0	0	1	0	0	0	0	1	1
1	1	0	1	1	0	0	0	0	1	1
1	1	1	0	1	1	0	0	1	0	0
1	1	1	1	1	1	0	0	1	0	0

Logical Gates and Circuits

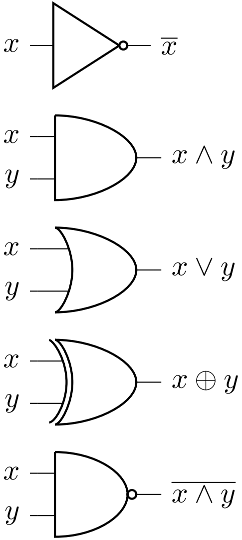
negation

conjunction

disjunction

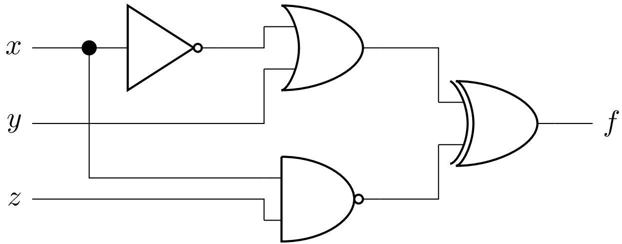
exclusive or (XOR)

NAND



Example: Building a Simple Digital Circuit

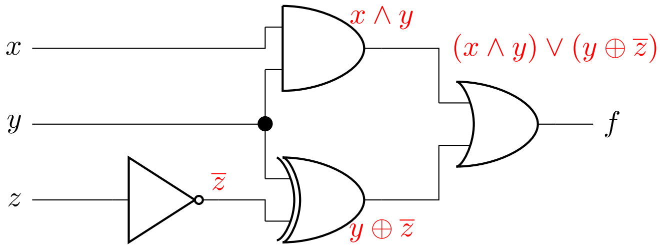
$f(x, y, z) = (\overline{x} \vee y) \oplus \overline{x \wedge z}$



Remark with 'simple' we mean 'no loops'
(i. e., no paths from an output of a gate to its own input)

Remark better idea of such circuits by 'playing' with Logisim
`http://www.cburch.com/logisim/`
freely available as `.jar` file
and in custom versions for Mac and Windows

Example: Reading a Simple Digital Circuit



$$f(x, y, z) = (x \wedge y) \vee (y \oplus \overline{z})$$

Summary & Take Home Message

Things to remember

- propositional logic and Boolean algebra
- Boolean operators
- truth tables
- digital gates and digital circuits

Take Home Message

- Boolean algebra is fundamental for computers,
  in particular computer hardware.
- Computations with Boolean algebra are tedious, not difficult.

Lecture feedback `http://onlineted.com`